MOTOROLA

FAX TRANSMITTAL SHEET

Motorola, Inc.
Intellectual Property Section
Law Department
101 Tournament Drive
Horsham, PA 19044

Telephone: 215-323-1907
Facsimile: 215-323-1300

37 Number of Pages (including this page)

Paper 2 of 2

Date:	January 27, 2006
To:	Examiner: Ponnoreay Pich Art Unit: 2135
Location:	United States Patent and Trademark Office
Fax No.:	571-273-8300
From:	Attorney: Robert P. Marley Reg. No. 32,914
Subject:	Serial No. 09/827,630 Filed: 4/06/2001 Docket No. D02316-04

NOTICE: This facsimile transmission may contain information that is confidential, privileged, or exempt from disclosure under applicable law. It is intended only for the person to whom it is addressed. Unauthorized use, disclosure, copying or distribution may expose you to legal liability. If you have received this transmission in error, please immediately notify us by telephone (collect) to arrange for return of the documents received and any copies made. Thank you.

MESSAGE:

Enclosed herewith, please find Amendment in Response to Non-Final Office Action dated July 27, 2005, Petition for Extension of Time, and Declaration Under 37 CFR 1.131for filing in the above-identified case.

PLEASE GIVE THESE PAPERS TO:

EXAMINER:	**Ponnoreay Pich**
GROUP ART UNIT:	2135
ATTORNEY DOCKET NO.:	D02316-04

The ACP can conceptually be extended to completely manage the entire downloadable memory space, so that no App exists, is loaded, is launched, or is allowed to continue running without the ACP having the unstoppable opportunity to erase it.

4. Related Considerations & Comments

4.1 Trust Levels for an OS

There are three obvious security worries for an OS. First, the OS could be manipulated in some way to evade a security function, or be tricked into making a different decision than normal for a security check. A historical example of this might be a security check implemented in a batch file under DOS. DOS allows the press of keys Control C to abort execution of a batch file, so it is possible to evade any check made in a batch file by carefully timed key presses.

The second problem is unsatisfactory design of the OS itself. This could occur if:

1. A pirate alters OS code to do something different that its designers put into it; or
2. The OS designers made a mistake during design; or
3. The OS designers intentionally designed around inconvenient functions, or placed "back doors" into the code for business or surreptitious employee reasons.

The third problem is intentional misdesign of the OS, either by a malicious employee of the OS vendor, or by conscious predatory intent of the OS vendor company.

These three cases[14] each mandate increasingly extreme and laborious measures to counter them:

1. The OS must be digitally signed, and this signature checked using the BIOS and ACP
2. The OS must be thoroughly reviewed at the source code level, and tested at the executable code level. This should preferably occur prior to release, by a disinterested party highly skilled in software design and Independent Validation and Verification (IVV) processes.
3. The thorough review of (2) above must include monitoring the creation of releasable code The actual process of compiling source code into executable code must be witnessed and very closely monitored by an independent party. The exact source code used in compilation must be stored outside the control of the OS vendor, then compilation observed, then executable code digitally signed and similarly stored. Under no circumstances can the OS vendor ever possess the digital signing key.
4. Various OS double checking roles for the ACP have been mentioned in this memo. The more the OS is of concern, the more weight should be given to implementing these

4.2 Securing Functions Outside the Application Layer

So far we have focused on Applications, which are the highest level software objects running in the 5000 They often interface with the user, perform GUI functions, etc. But other software objects in

[14] Note that some OS vendors have been known to have very buggy software, to have frequently misdesigned software with unintended function, and to be predatory against other companies

the 5000 are also the subject of security. Items like a TCPIP stack, telephone or cable modem driver, or other lower level software can be secured for sale through A&A.

This is achievable through the same mechanisms as securing Apps. The CA system and its ACP have no awareness of OS or software details, only that the A&A process moves certain specific data around for processing in specific ways. It makes no difference to A&A if the segment of data being Authorized and Authenticated comprises any of the following:

- Applications
- Protocol stacks
- Hardware drivers
- Data files
- A Java Virtual Machine
- A Java applet

All can be secured by the same CA system mechanisms and cryptography. Data is data, and the nature of the data is transparent to the CA system. But impact outside the CA system is huge

Each and every software object to be subjected to A&A must have an ECDS. The functions of A&A described in this memo must be added to BIOS and OS, and to any software that manipulates other software. For example, it must not be possible to launch the equivalent of a small OS as an App, as that App-OS-on-top-of-an-App can itself launch Apps that are not subject to A&A. (This is Java!)

4.3 A Warning Regarding the Java Virtual Machine

The Java Virtual Machine (JVM) is an Application level artifice that serves as an OS of sorts all by itself. Java Applets run on the JVM, and these Apps are as significant to the network operator's business as are non-Java Apps. Java Applets must be secured using the exact same techniques, and to the exact same security level as regular Apps, without exception.

For every paragraph in this memo that has the words "Operating System" or "OS", a mirror paragraph stating Java Virtual Machine can be written. Securing the Java Applets is just as paramount, and if they are ignored, then there is little point in securing regular Applications.

This is true even if a JVM is not present in the original 5000 product launch. When a JVM is added at a later time (perhaps as a downloaded APPV App), then it must be a JVM designed to work with the ACP at the security level selected for the entire 5000 system.

5. Summary

Though securing Applications and other software objects is easy to conceive for a CA system, this is absolutely not the case for the rest of the system!! In fact, the design impact of securing any software object whatsoever as described in this memo is substantial. The precepts and paradigms of current software engineering practice view the security techniques discussed herein as complete and horrible anathema. It must not be underestimated how mind-bending a change this represents. Radical new approaches will naturally be met with resistance, and will encounter unforeseen problems, and will require substantial development effort and time. Consideration of this downside and its risks is important!

A complete implementation is also paramount. Any single omission creates an Achilles Heel, and suggests that an All-or-Nothing philosophy be applied. Unless the protections appropriate to a specific chosen Application Security Level can be done completely and without any exception whatsoever, it is recommended that they not be undertaken. If Apps are secured, for example, but the OS is not, then failure of App security is predicted.